

Extension operations on sets of leaf-labelled trees

by

David J. Bryant and M.A. Steel

*Department of Mathematics and Statistics
University of Canterbury, Christchurch, New Zealand.*

No. 118

November, 1994

Extension operations on sets of leaf-labelled trees

David J. Bryant* and Mike A. Steel†

Department of Mathematics & Statistics
University of Canterbury, Christchurch,
New Zealand

November 9, 1994

Abstract

A fundamental problem in classification is how to combine collections of trees having overlapping sets of leaves. The requirement that such a collection of trees is realized by at least one parent tree determines uniquely some additional subtrees not in the original collection. We analyse the ‘rules’ that arise in this way by defining a closure operator for sets of trees. In particular we show that there exist rules of arbitrarily high order which cannot be reduced to repeated application of lower-order rules.

Keywords: Phylogenetic trees, subtrees, graphs, compatibility, closure.

*Email: djb@math.canterbury.ac.nz

†Email: mas@math.canterbury.ac.nz

1 Preliminaries

Introduction

Trees with labelled leaves are useful models for representing evolutionary relationships, particularly in biology (where they are called phylogenetic trees). The wider availability of genetic sequence data, and the use of tree-building programs such as PAUP, PHYLIP and MACCLADE, has led to a substantial increase in the size and number of phylogenetic trees. This trend has heightened the relevance of the *generalized tree compatibility problem*: determining whether a collection of phylogenetic trees on overlapping sets of taxa can be combined into one all-inclusive tree. Any ‘divide and conquer’ technique for large classifications encounters this problem, as would any attempt to incorporate the many existing phylogenies into new phylogenetic trees. Tree compatibility can be efficiently determined when all the input trees are either all rooted or have a leaf in common [2, 6, 12]. If the trees have the same leaf sets then compatibility can be determined in linear time [18]. However the general problem for unrooted trees is NP-complete [14].

A further problem in combining phylogenetic trees is that any tree constructed might be only one among a multitude of possible trees, each of which is well supported by the data. This situation arose, for example, in work by Cann *et al.* [5] involving the evolution of human mitochondrial DNA. Maddison [11] argues that the tree used to assert the African origin of human mtDNA is only one among many equally plausible trees, some which even support an Asian origin. In practice there are often thousands of suitable trees consistent with any given data set. This reflects the exponentially large number of possible phylogenetic trees.

Our approach is to break the initial collection of trees into an equivalent set of binary trees, each with three leaves (rooted triples) or four leaves (quartets). In this way, many of the original problems involving phylogenetic trees can be converted into equivalent problems involving these sets.

Dekker [7] investigated the use of quartets to construct a form of predicate calculus. Unlike standard predicate calculus there would be three possible logical values, corresponding to the three possible quartets on a set of four leaves. This approach led to a number of inference rules: a set Q of quartets ‘implies’ another quartet q if every tree compatible with Q is also compatible with q . In this way we can deduce new phylogenetic information that is not explicitly present in the initial data set. The same principles apply to sets of rooted triples.

We introduce *closed sets* — sets of quartets or rooted triples which cannot be extended by applying inference rules. The associated closure operator, which replaces a set by the minimal closed set containing it, has a number of attractive properties, especially when applied to sets of rooted triples. The closure of a set contains the

triple/quartet information that can be directly inferred from that set.

Despite the fact that inference rules are defined in such a simple manner, the set of all inference rules exhibits a remarkable complexity. In particular, there is no finite list of quartet or rooted triple rules that generates all other rules through repeated application. This result was first conjectured by Dekker [7], and we prove it by using a graph theoretic approach to the study of closed sets. An outline of the paper is as follows:

- In the remainder of this section we define phylogenetic trees and compatibility, giving a brief survey of related concepts in the literature. We characterize compatibility in terms of quartets and rooted triples, and discuss when a collection of subtrees defines a unique tree.
- Section 2 introduces compatibility rules for quartet sets and prove a number of related properties.
- Section 3 examines sets of rooted triples, and present a new graphical characterization of consistency and closure.
- In Section 4 we use this graphical representation to prove that there are rules of any order that cannot be derived from rules of lesser order. The result is proved first for rooted triples and then extended to quartets.

Phylogenetic Trees

An **unrooted (phylogenetic) tree** is an acyclic connected graph with no vertices of degree two, and with each leaf (vertex of degree one) labelled uniquely. This corresponds to a phylogenetic tree in [8, 14], to a semilabelled tree in [15], an S-labelled tree in [3] and a fully resolved tree structure in [4].

Much of the work in classification involves **rooted** phylogenetic trees. One internal vertex, which in this paper will always be labelled ρ , is distinguished and called the ‘root’. For example, the ancestral element of a cladogram is often taken as the root.

In a **binary unrooted phylogenetic tree** every internal (i.e. non-leaf) vertex has degree three. This is called a non-degenerate tree structure in [4]. In a **binary rooted phylogenetic tree**, all internal vertices have degree three, except the root which has degree two.

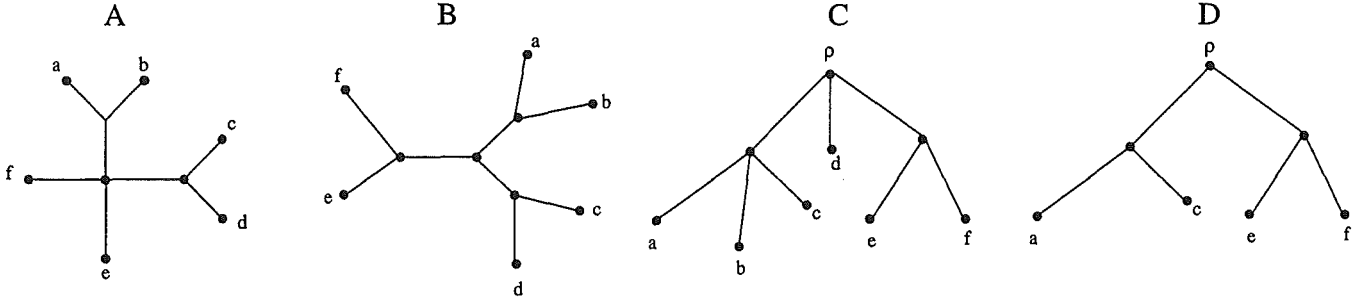


Figure 1 : Four examples of *phylogenetic trees*. A and B are *unrooted*. C and D are *rooted*.
B and D are *binary*.

Given any tree T , let $\mathcal{L}(T)$ be the leaf set of T . If \mathcal{T} is a set of trees, let $\mathcal{L}(\mathcal{T})$ be the union of the leaf sets of the trees in \mathcal{T} .

Sometimes the internal vertices of a phylogenetic tree are labelled, or a vertex might have more than one label [7, 8]. (These trees are also called ‘S-labelled Trees’ [18], or ‘Tree Structures’ [4]). Eldredge and Cracraft discuss the various merits of each tree type and observe that all of the classification information contained in a tree with labelled internal vertices can be represented in a tree with only the leaves labelled [9, pg 211ff].

Rooted phylogenetic trees are sometimes displayed with a vertical axis representing the time each branching point occurred. These diagrams are called dendrograms. In the present paper we are only concerned with the underlying branching tree structure.

Compatibility

Let T be a rooted or unrooted phylogenetic tree. A **contraction** of T is obtained by removing an internal edge and identifying its endpoints.

Let A be a subset of the leaf set $\mathcal{L}(T)$ of T . Remove all the leaves of T not in A , together with their adjoining edges. Delete any internal vertices that have only two remaining neighbours and identify their two incident edges. The resulting tree is called the **subtree of T induced by A** and is denoted $T|_A$ [14].

For example, consider Figure 1. Tree D is an induced subtree of tree C. Tree A is obtained from tree B by a contraction of the horizontal edge, but it is not an induced subtree of B.

We say that a tree T is **compatible with** a tree S if S can be obtained by contractions of an induced subtree of T (or equivalently, if S is an induced subtree of a contraction of T). We denote this partial order by $S \leq T$. A tree T^* is compatible

with a set of trees $\mathcal{T} = \{T_1, \dots, T_k\}$ if T^* is compatible with each T_i , in which case we say that \mathcal{T} is **consistent**. This definition of compatibility corresponds to ‘weak’ compatibility in [12]. The terms consistent and compatible are sometimes interchanged in the literature [12, 14].

The underlying assumption made when choosing this type of compatibility is that the tree structures we are trying to model are binary. Hence a non-binary tree corresponds to incomplete knowledge. It is the branching information that we are most interested in. In cladograms, the branching structure determines the nesting of the sets of taxa. Our definition of compatibility corresponds to one tree containing all the clustering information of the other tree (Theorem 1 (1) of [8]), or alternatively to one tree containing all the nested set information of the other tree (Corollary 1).

There are, however, several versions of compatibility in common use. For example, [3, 6, 12] do not incorporate contraction into their definitions of compatibility.

Quartets and Rooted Triples

A useful way to analyse trees and sets of trees is in terms of their smallest phylogenetically informative subtrees — rooted triples for rooted trees, and quartets for unrooted trees.

Definitions

1. A **quartet** is an unrooted binary tree with four leaves. The quartet with two pairs of leaves $\{a, b\}$ and $\{c, d\}$ connected by an internal edge is denoted $ab|cd$. A **rooted triple** is a rooted binary tree with three leaves. The rooted triple with a pair of leaves $\{a, b\}$ connected to the third leaf c via the root is denoted $ab|c$. Adams [1] uses the term ‘triad’ for rooted triples.

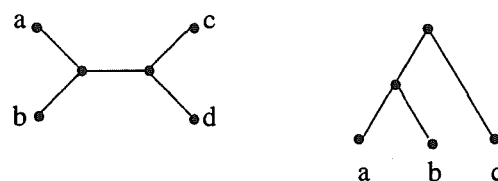


Figure 2 : The quartet $ab|cd$ and the rooted triple $ab|c$.

2. If Q is a set of quartets, then the **span** of Q , or $\langle Q \rangle$ is the set of unrooted trees that are compatible with Q and have leaf sets $\mathcal{L}(Q)$. Similarly, if R is a set of rooted triples, then the **span** of R , or $\langle R \rangle$ is the set of rooted trees

that are compatible with R and have leaf sets $\mathcal{L}(R)$. The algorithm ALLTREES of [12] and the algorithm SUPERB [6] can both be used to construct the span of a set of rooted triples.

3. Let $r(T)$ denote the set of rooted triples that are induced subtrees of a rooted tree T , and let $q(T)$ denote the set of quartets that are induced subtrees of an unrooted tree T . The set $r(T)$ is called the **rooted triple set** of T , and $q(T)$ is called the **quartet set** of T .
4. Given a phylogenetic tree T , deleting an edge gives two smaller subtrees and thereby a partition of the leaf set of T into two non-empty subsets. Such a partition is called a **split** of T .

The following theorem extends a result of [8] giving a characterization of compatibility and the partial order \trianglelefteq .

Theorem 1 *Let S and T be unrooted phylogenetic trees. T is compatible with S , that is $S \trianglelefteq T$, if and only if $q(S) \subseteq q(T)$ and $\mathcal{L}(S) \subseteq \mathcal{L}(T)$. Similarly, let S and T be rooted phylogenetic trees. $S \trianglelefteq T$ if and only if $r(S) \subseteq r(T)$ and $\mathcal{L}(S) \subseteq \mathcal{L}(T)$.*

Proof

Suppose first that $S \trianglelefteq T$. If $ab|cd \in q(S)$ then $ab|cd \trianglelefteq S$. Since \trianglelefteq is transitive, we have that $ab|cd \trianglelefteq T$ and so $ab|cd \in q(T)$. Of course if T is compatible with S then we also have $\mathcal{L}(S) \subseteq \mathcal{L}(T)$.

Conversely, suppose that $q(S) \subseteq q(T)$ and $\mathcal{L}(S) \subseteq \mathcal{L}(T)$. If we can prove that the subtree of T induced by $\mathcal{L}(S)$ is compatible with S , then $S \trianglelefteq T$. Therefore we assume that $\mathcal{L}(T) = \mathcal{L}(S)$. We will show that the set of splits of S is contained in the set of splits of T so that the result follows from Theorem 1,(1) in [8].

Let $(\eta, \bar{\eta})$ be a split of S . Then $ab|cd \in q(S)$ for all $a, b \in \eta$ and $c, d \in \bar{\eta}$ [4]. Since $q(S) \subseteq q(T)$, $ab|cd \in q(T)$ for all $a, b \in \eta$ and $c, d \in \bar{\eta}$. Hence $(\eta, \bar{\eta})$ is a split of T .

An analogous argument applies for the rooted case. \square

Note that if $q(S) \subseteq q(T)$ then $\mathcal{L}(S) \subseteq \mathcal{L}(T)$, unless S is a fan-like tree with no internal edges, in which case $q(S) = \emptyset$.

Adams [1] defines a partial order $<_T$ on sets of leaves in a rooted tree T . Let X, Y be subsets of $\mathcal{L}(T)$. Then $X <_T Y$ if the most recent common ancestor of X is a descendent of the most recent common ancestor of Y . We say that X *nects* in Y . The partial order defines the tree uniquely. We show that compatibility corresponds to one tree containing all the nesting information of the other tree.

Corollary 1 *Let S and T be two rooted phylogenetic trees. $S \trianglelefteq T$ if and only if $A <_S B$ implies $A <_T B$.*

Proof

Note that $\{a, b\} <_T \{a, b, c\}$ if and only if $ab|c \in r(T)$. Suppose that $A <_S B$ implies $A <_T B$. Clearly $\mathcal{L}(S) \subseteq \mathcal{L}(T)$. If $ab|c \in r(S)$ then $\{a, b\} <_S \{a, b, c\}$ so $\{a, b\} <_T \{a, b, c\}$ and $ab|c \in r(T)$. By Theorem 1, $S \trianglelefteq T$.

Conversely suppose that $S \trianglelefteq T$. If $A <_S B$ then the most recent common ancestor of A is a descendant of most recent common ancestor of B . This will still be true if we add leaves and expand contracted vertices. Hence $A <_T B$. \square

Consensus Trees

A rooted tree can be defined in terms of its nesting partial order, in terms of its rooted triples, or in terms of its splits. To represent the consensus information shared by a number of rooted trees, a desire would be to preserve the nestings, rooted triples, or splits common to all the trees. Adams [1] observed that trees tend to have more nesting information in common than can be obtained from the intersection of their rooted triple sets, and the Adams consensus tree is constructed from this shared nesting information. In contrast, the strict consensus subtree, which is constructed from the splits common to all the trees, contains less information than the intersection of the rooted triple sets. This decrease in shared information, from nestings to rooted triple sets to splits, is discussed in detail in [1] and [16]. A simple comparison between the three approaches is obtained by studying rooted triple sets, as follows.

Proposition 1 *Let $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ be a collection of rooted trees with the same leaf set. If T_{AC} is the Adams consensus tree of \mathcal{T} , T_{SC} is the strict consensus tree of \mathcal{T} , and $R = \bigcap_{T \in \mathcal{T}} r(T)$ then*

$$r(T_{SC}) \subseteq R \subseteq r(T_{AC}).$$

Proof

We have $T_{SC} \trianglelefteq T$, $\forall T \in \mathcal{T}$. If $ab|c \in r(T_{SC})$, then $ab|c \in r(T)$ for all $T \in \mathcal{T}$, so $ab|c \in R$.

If $ab|c \in R$ then $\{a, b\} <_T \{a, b, c\}$ for all $T \in \mathcal{T}$, so $\{a, b\} <_{T_{AC}} \{a, b, c\}$. \square

The algorithm OneTree

A rooted tree T satisfies the constraint $(a, b) < (c, d)$ if the most recent common ancestor of a and b is a descendent of the most recent common ancestor of c and d , that is, if $\{a, b\} <_T \{c, d\}$. Aho *et al.* [2] present an algorithm BUILD that returns a tree compatible with a set of constraints whenever such a tree exists. The set of all possible constraints that can be obtained from any one tree is characterized in [12].

Aho *et al.*'s algorithm has been extended and modified [6, 12]. Constantinescu and Sankoff [6] present an algorithm SUPERB that takes a set of constraints and returns all of the binary trees compatible with them, if any such trees exist. Ng and Wormald [12] give two tree construction algorithms ONETREE and ALLTREES. These take rooted triples and k -leaved fan trees as input, where a tree T is defined to be compatible with a fan tree S if S is an induced subtree of T . ONETREE constructs a single compatible tree and ALLTREES lists all compatible trees.

The algorithm given below is a simplification of ONETREE. It does not handle fan trees. In addition, instead of constructing the blocks of a partition at each iteration, we construct a graph and consider its components. This graphical approach was used in [2] to show that their algorithm has $O(m^2)$ complexity when applied to a set of m rooted triples.

ONETREE(R, A, v, T)

Input: set R of rooted triples,
 non-empty set $A = \{a_1, \dots, a_n\}$ containing the leaves of R ,
 vertex v .

Output: tree T with root v .

1. If $n = 1$, set $T = v$ with label a_1 and return.
 If $n = 2$, create T by attaching two new vertices to v , label them a_1 and a_2 and return.
2. Create a graph G with vertices A and an edge between a and b if there is an element $c \in A$ such that $ab|c \in R$.
3. If G has only one component then set $T = \emptyset$ and return.
4. For each component A_i of G , create a vertex v_i :
 set $R_i :=$ the set of rooted triples in R with leaves in A_i , and call
 ONETREE(R_i, A_i, v_i, T_i).
 If $T_i = \emptyset$ then set $T = \emptyset$ and return.
 Otherwise, add T_i and the edge (v, v_i) to T .

The algorithm has complexity $O(n^4)$, where n is the number of leaves in the input set of rooted triples.

Defining a tree by collections of subtrees

Given a collection of input trees, a natural question is whether there is *exactly one* parent tree compatible with each input tree, in which case we say that the input

defines the parent tree. When considering sets of quartets and rooted triples the question becomes when does the span $\langle X \rangle$ of a consistent set contain only one tree?

One immediate observation is that if $\langle X \rangle$ contains a non-binary tree T , then $\langle X \rangle$ also contains all the binary trees that can be contracted to T . Note also that any binary tree T is defined by $q(T)$ [4].

Necessary and sufficient conditions for defining a rooted tree by rooted triples are presented in [14]. Given an edge e of a rooted tree S , and a rooted triple $ab|c \in r(S)$, we say that $ab|c$ **distinguishes** e if the path from a to c in S intersects the path from b to the root of S only on the edge e . It was shown in [14] that a set of rooted triples R defines a unique binary rooted tree T if and only if each edge in T is distinguished by a rooted triple in R . We can also characterize when a set of triples defines a tree using the algorithm ONETREE.

Proposition 2 *The algorithm ONETREE returns a binary tree T when applied to a set of rooted triples R if and only if R defines T .*

Proof

Suppose that the algorithm applied to R returns a binary tree T . The algorithm constructs a graph G with two components. Each component contains the leaves of one of the two subtrees of T branching off the root of T . Let T' be any rooted tree with leaf set $\mathcal{L}(R)$ that is compatible with R . By Lemma 1 of [2], the two components of G are wholly contained in subtrees branching off the root of T' . Hence the subtrees that branch off the root of T' have the same leaves as the subtrees branching off the root of T . Now the algorithm ONETREE recurses on these subtrees of T and so every subtree of T has the same leaves as some subtree of T' . It follows that $T = T'$.

Conversely if R defines T then the tree returned by ONETREE would have to equal T . If T was not binary, then any tree that contracts to give T would also be compatible with R , a contradiction. \square

Note that Proposition 2 can also be proved by referring to the algorithm ALL-TREES of [12].

Let T be an unrooted tree and let e be an edge of T . We say that the quartet $ab|cd$ **distinguishes** e if the path from a to c in T , and the path from b to d in T intersect only on the edge e . If a set of quartets Q defines a tree T (that is, $\langle Q \rangle = \{T\}$) then, by Proposition 6 in [14], every edge of T is distinguished by a quartet of Q . Hence if Q defines T and T has n leaves then $|Q| \geq n - 3$, the number of internal edges of T [14, 13]. This lower bound can be realized for every unrooted tree, by a construction given in [14].

We can generalize this result from sets of quartets to collections of trees.

Proposition 3 *If $\{T_1, T_2, \dots, T_k\}$ is a set of trees such that $\langle \{T_1, T_2, \dots, T_k\} \rangle = \{T\}$ for some binary tree T with n leaves, then*

$$\sum_{i=1}^k (n_i - 3) \geq n - 3 \quad (1)$$

where n_i is the number of leaves in T_i .

Proof

For each tree T_i let S_i be the binary subtree of T induced by the leaves of T_i so that $S_i = T|_{\mathcal{L}(T_i)}$. Hence for each i , $q(T_i) \subseteq q(S_i)$ so the trees $\{S_1, S_2, \dots, S_k\}$ define T . Using the construction of [14], let Q_i be a set of $n_i - 3$ quartets that defines S_i , $i = 1, 2, 3, \dots, k$. Hence $Q_1 \cup Q_2 \cup \dots \cup Q_k$ is a set of quartets that defines T . It takes at least $n - 3$ quartets to define T , so

$$\begin{aligned} n - 3 &\leq |Q_1 \cup Q_2 \cup \dots \cup Q_k| \\ &\leq |Q_1| + |Q_2| + \dots + |Q_k| \\ &= \sum_{i=1}^k (n_i - 3) \end{aligned}$$

as required.

Note that even when the inequality (1) does hold for a particular choice of n_1, n_2, \dots, n_k and n it does not necessarily follow that there exist trees T_i with n_i leaves, ($i = 1, \dots, k$), which define a given binary tree T with n leaves. The smallest counterexample is given by any binary tree with $n = 9$ leaves that has a vertex with 3-fold symmetry. Such a tree cannot be defined by two subtrees each with 6 leaves.

The tree T will not, in practice, be known in advance. In this case, Warnow [17] showed that we only need to examine $O(n \log n)$ of the quartets in $q(T)$ before we can uniquely determine T .

2 Closed Sets - Quartets

Any question relating to the compatibility of unrooted trees can be converted into a question about sets of quartets. In this section we study sets of quartets and introduce the concept of closed sets of quartets. Closed sets arise in two different contexts: firstly in terms of the inference rules of [7], and secondly as the intersection of quartet sets of trees.

Inference Rules for Quartets

Let T be an unknown unrooted phylogenetic tree. Given a subset of $q(T)$, it is often possible to deduce additional quartets of $q(T)$. For example:

1. If $ab|cd, ab|ce \in q(T)$ then $ab|de \in q(T)$. [7]
2. (Dees) If $ab|cd, ac|de \in q(T)$ then $ab|ce \in q(T)$. [7, 14]
3. If $ab|cd, ab|ef, ce|df \in q(T)$ then $ab|df \in q(T)$. [7]

We generalize these results by defining abstract inference rules. A **rule** is a statement of the form: “If $Q \subset q(T)$ then $ab|cd \in q(T)$ ” and is denoted $Q \vdash ab|cd$. Hence $Q \vdash ab|cd$ is true if every tree compatible with a particular set of quartets Q is always compatible with the quartet $ab|cd$. Given a consistent set of quartets Q , define

$$\overline{Q} := \bigcap_{T \in \langle Q \rangle} q(T).$$

Thus $Q \vdash ab|cd$ is a rule if and only if $ab|cd \in \overline{Q}$. The set \overline{Q} is called the **closure** of Q . A set Q is **closed** if every rule $Q \vdash ab|cd$ implies that $ab|cd \in Q$. The **order** of the rule $Q \vdash ab|cd$ is equal to the cardinality of Q .

We present a number of basic properties of closed sets and the closure operator, all of which follow immediately from the definitions of closure and closed sets.

Proposition 4 *Let X, Y be consistent sets of quartets.*

1. \overline{X} is the minimal closed set containing X .
2. $\overline{X} = \overline{(\overline{X})}$.
3. If $X \subseteq Y$ then $\overline{X} \subseteq \overline{Y}$.
4. X is closed if and only if $X = \overline{X}$.

5. If X and Y are closed sets then $X \cap Y$ is also closed.
6. T is compatible with X if and only if T is compatible with \overline{X} .
7. $\langle X \rangle = \langle Y \rangle$ if and only if $\overline{X} = \overline{Y}$.

Closed sets and quartet sets of trees

The definition of closure suggests a link between closed sets and quartet sets of trees. In fact, the quartet sets of binary trees are the maximal closed sets, and all other closed sets can be written as the intersection of them.

Proposition 5 *X is closed if and only if $X = q(T_1) \cap q(T_2) \cap \dots \cap q(T_k)$ for some trees T_1, T_2, \dots, T_k . Furthermore we can assume that T_1, T_2, \dots, T_k are binary.*

Proof

Clearly, if T is a tree, then $\overline{q(T)} = q(T)$, so $q(T)$ is closed. If $X = q(T_1) \cap q(T_2) \cap \dots \cap q(T_k)$ for some trees T_1, T_2, \dots, T_k then X is closed, by Proposition 4 (5).

Conversely if X is closed then $X = \overline{X}$ which is, by definition, the intersection of the quartet sets of all the trees compatible with X . We can restrict our attention to binary trees because the quartet set of $q(T)$ of any non-binary tree T equals the intersection of the quartet sets of the binary trees compatible with T . \square

Proposition 6 *If X and Y are consistent sets of quartets and $\mathcal{L}(X) \cap \mathcal{L}(Y) = \emptyset$ then $X \cup Y$ is consistent and $\overline{X \cup Y} = \overline{X} \cup \overline{Y}$.*

Proof

First assume that there are unrooted trees T_1 and T_2 such that $X = q(T_1)$ and $Y = q(T_2)$. We can combine T_1 and T_2 into a single tree by identifying an internal vertex of T_1 with an internal vertex of T_2 . Any tree thereby constructed is compatible with $q(T_1) \cup q(T_2)$, so $X \cup Y$ is consistent. Each different pair of internal vertices gives rise to a different tree. Let \mathcal{T} be the collection of all these trees. Then

$$q(T_1) \cup q(T_2) = \bigcap_{T \in \mathcal{T}} q(T)$$

so $X \cup Y = q(T_1) \cup q(T_2)$ is closed by Proposition 5.

Suppose that X and Y are any two consistent sets. By the definition of closure,

$$\overline{X} = \bigcap_{T_1 \in \langle X \rangle} q(T_1) \text{ and } \overline{Y} = \bigcap_{T_2 \in \langle Y \rangle} q(T_2).$$

Hence

$$\begin{aligned}\overline{X \cup Y} &= \left(\bigcap_{T_1 \in \langle X \rangle} q(T_1) \right) \cup \left(\bigcap_{T_2 \in \langle Y \rangle} q(T_2) \right) \\ &= \bigcap_{T_1 \in \langle X \rangle, T_2 \in \langle Y \rangle} (q(T_1) \cup q(T_2))\end{aligned}$$

which is closed by the first part and Proposition 4 (5). Now $\overline{X \cup Y} \subseteq \overline{X \cup Y}$, and $\overline{X \cup Y}$ is the minimal closed set containing $X \cup Y$. Since $\overline{X \cup Y}$ is closed, it follows that $\overline{X \cup Y} = \overline{X \cup Y}$, as required. \square

Some applications of closed sets

(1) Let \mathcal{T} be collection of trees T_1, T_2, \dots, T_k and let $Q = q(T_1) \cap q(T_2) \cap \dots \cap q(T_k)$. The set of quartets Q is often taken to be the consensus information shared by all the trees in \mathcal{T} . By Proposition 5 any such set is closed. As well, if S is any consensus tree such that $S \leq T_i$, $i = 1, \dots, k$, then $q(S) \subseteq Q$. Note that other consensus methods are in use. Some consensus trees, like Adams consensus tree [1], preserve more information than is contained in the intersection of quartet (or rooted triple) sets.

(2) Let Q be a set of quartets or trees and let $n = |\mathcal{L}(Q)|$. The number of trees compatible with Q can be exponentially large with respect to n , so it is often impractical to list every possible tree. Instead we could use the closed set \overline{Q} to represent the set of possible trees. The set \overline{Q} contains exactly those quartets that can be directly deduced from Q .

(3) Another advantage of using closed sets to process phylogenetic information is that the collection of closed subsets of a closed set, partially ordered by inclusion, forms a complete lattice, with

$$\bigwedge_i X_i = \bigcap_i X_i \quad \text{and} \quad \bigvee_i X_i = \overline{\bigcup_i X_i}$$

for closed sets X_i . In contrast, the set of trees partially ordered by compatibility (\leq) has no well defined meet and join, even when the trees are consistent.

(4) In section 1 we discussed the question “When does a set of quartets define a tree?” An answer is provided by the closure operator. If $\langle X \rangle$ consists of just one tree T then $\overline{X} = q(T)$. Conversely if $\overline{X} = q(T)$ for some binary tree T then $\langle X \rangle = \langle \overline{X} \rangle = \langle q(T) \rangle = \{T\}$. Therefore a set of quartets X defines a tree T if

and only if T is binary and $\overline{X} = q(T)$.

(5) Phylogenetic information is often given by sets of characters. Each character gives a partition of the set of species, that is, a partition of the leaf set. Given a partition $A_1|A_2|\dots|A_k$ define the set of quartets

$$q(A_1|A_2|\dots|A_k) := \{wx|yz : w, x \in A_i; y, z \in A_j; i \neq j\}.$$

There is a corresponding notion of compatibility with partitions [8, 10, 17]. It can be shown that a tree T is compatible with the partition $A_1|A_2|\dots|A_k$ if and only if $q(A_1|A_2|\dots|A_k) \subseteq q(T)$. Hence every tree compatible with X is compatible with the partition $A_1|A_2|\dots|A_k$ if and only if $q(A_1|A_2|\dots|A_k) \subseteq \overline{X}$. The inference rules of [7] involving partitions can therefore be reduced to inference rules involving quartets, giving additional motivation for studying sets of quartets.

We prove that for any partition $A_1|A_2|\dots|A_k$, the set of quartets $q(A_1|A_2|\dots|A_k)$ is closed. In order to do so we consider quartet sets of graphs that are not necessarily trees.

Lemma 1 *Let G be any connected graph and let L be a set of labelled vertices in G . Define*

$$q(G) := \left\{ ab|cd : \begin{array}{l} a, b, c \text{ and } d \text{ are distinct elements of } L \\ \text{no path from } a \text{ to } b \text{ intersects a path from } c \text{ to } d \end{array} \right\}$$

Then $q(G)$ is consistent and closed.

Proof

We can assume that every vertex in G is on a path between two elements of L , since removing these vertices does not change the set $q(G)$.

Consider first the case when G is acyclic. Suppose that G has an internal vertex labelled a . If we attach a new leaf adjacent to this vertex and transfer the label a from the internal vertex to the leaf a , then $q(G)$ will not change. Repeat this procedure until all labelled vertices of G are leaves. If we now delete those vertices that have only two remaining adjacent vertices and identify their incident edges then we obtain a phylogenetic tree T with $q(T) = q(G)$. Hence $q(G)$ is consistent and closed.

Suppose now that G is not acyclic. Let τ be a spanning tree of G . Since τ is acyclic so $q(\tau)$ is the quartet set of some phylogenetic tree. If $ab|cd \in q(G)$ then no path from a to b intersects a path from c to d in G and because τ is a subgraph of G , the same applies for τ . Hence $q(G) \subseteq q(\tau)$ and so $q(G)$ is consistent.

Let \mathcal{T} be the collection of spanning trees of G . We will show that

$$q(G) = \bigcap_{\tau \in \mathcal{T}} q(\tau)$$

and therefore $q(G)$ is closed by Proposition 5. If $ab|cd \notin q(G)$ then there is a path P_1 from a to b that intersects a path P_2 from c to d . Let x be the *first* vertex on the path P_2 that is also on the path P_1 . Let y be the *last* vertex on the path P_2 that is also on the path P_1 . Construct the subgraph of G containing all of P_1 , the part of P_2 going from c to x and the part of P_2 going from y to d . This subgraph is an independent set of the graph matroid so can be extended to a spanning tree τ of G for which $ab|cd \notin q(\tau)$ [19, chpt 1]. Hence $ab|cd \notin \bigcap_{\tau \in \mathcal{T}} q(\tau)$. \square

Unfortunately not every consistent closed set is $q(G)$ for some graph G , a counterexample being the set $\{ab|cd, ab|ef\}$.

Proposition 7 *If $A_1|A_2|\dots|A_k$ is any partition then $q(A_1|A_2|\dots|A_k)$ is consistent and closed.*

Consider the graph G of Figure 3. Clearly $q(G) = q(A_1|A_2|\dots|A_k)$ so, by Lemma 1, the set of quartets is consistent and closed.

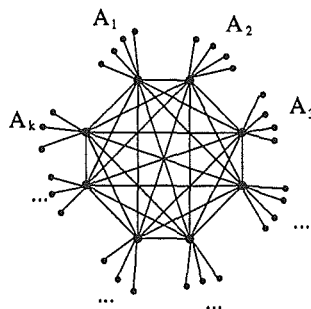


Figure 3 : The graph G with the same quartet set as $A_1|A_2|\dots|A_k$.

3 Closed Sets: Rooted Triples

Determining consistency is much easier for sets of rooted triples than for sets of quartets. The former can be checked in polynomial time [2] while the later problem is NP-complete [14]. The differences between the two cases are reflected by a number of properties that hold for sets of rooted triples, but not for sets of quartets. We begin by presenting a graphical characterization of consistency for sets of rooted triples.

Graphical representation of rooted triples: $R, S \longrightarrow [R, S]$

Let R be a set of triples and let S be a subset of $\mathcal{L}(R)$. We define an edge-labelled graph $[R, S]$ as follows. Take the vertices of the graph to be the elements of S . Add an edge between two vertices a and b if there are any triples in R of the form $ab|c$ where $a, b, c \in S$. Label each edge (a, b) with the set of leaves $\{x : ab|x \in R, x \in S\}$.

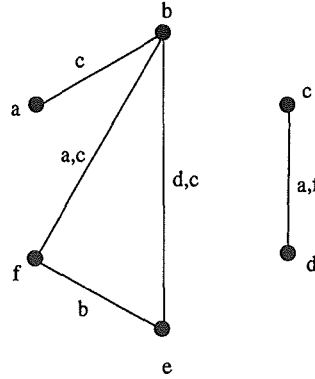


Figure 4 : The graph $[R, S]$ for $R = \{ab|c, be|d, be|c, af|g, fe|b, bf|a, bf|c, cd|a, cd|f, cg|b\}$ and $S = \{a, b, c, d, e, f\}$. The triples in R with a leaf g are ignored when constructing the graph because $g \notin S$.

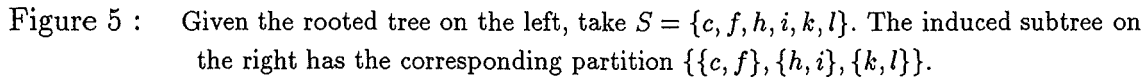
Every label on every edge of the graph represents a unique triple in R . Hence removing triples from R will corresponds to removing labels and, perhaps, edges from $[R, S]$. We summarize this observation as follows.

Proposition 8 *If R' is a subset of R , and S is a set of leaves, then $[R', S]$ is a subgraph of $[R, S]$. Consequently, if T is any rooted tree consistent with R , then $[R, S]$ is a subgraph of $[r(T), S]$.*

This graphical construction is closely related to the algorithm ONETREE. The algorithm returns a tree if and only if the input set of rooted triples is consistent,

Theorem 2 *A set of rooted triples R with leaf set L is consistent if and only if for each subset $S \subseteq L$ with at least three elements, the graph $[R, S]$ is disconnected.*

If R is consistent then there is a tree T such that $R \subseteq r(T)$. Let $S \subseteq L, |S| > 1$, and consider the subtree $T|_S$, which has a greatest element, say M . Each direct descendent x of M determines a subset of S given by those leaves that are descendants of x . The collection of these subsets partitions S into two or more blocks (see Figure 5).



Conversely, suppose that R is inconsistent. The algorithm ONETREE will return a null tree when applied to R . The algorithm acts recursively on different subsets A of $\mathcal{L}(R)$, and constructs a graph with the same vertices and edges as the graph $[R, A]$. It only returns a null tree when for some leaf set A , $|A| \geq 3$, the graph is connected. \square

Let T be a rooted tree compatible with R . If any of a , b or c is not in $\mathcal{L}(R)$ then

we can always add these extra leaves to T to give a tree compatible with $R \cup \{ab|c\}$ contradicting the inconsistency of $R \cup \{ab|c\}$.

By Theorem 2 there is a set S with $|S| \geq 3$ such that the graph $[R \cup \{ab|c\}, S]$ is connected. The graph $[R \cup \{ab|c\}, S]$ is the same as the graph $[R, S]$ with one extra edge connecting a and b and labelled by c . Hence $[R, S]$ has at most two components, and since R is consistent, the graph must have exactly two components. Adding the edge (a, b) gives a connected graph, so a and b must be in different components of $[R, S]$. \square

Closed sets of rooted triples

Closed sets and inference rules of rooted triples are defined in the same way as for quartets. The closure of a consistent set R is

$$\overline{R} = \bigcap_{T \in \langle R \rangle} r(T).$$

All of Proposition 4 holds for closed sets of rooted triples, and every closed set R of rooted triples can be written $R = r(T_1) \cap r(T_2) \cap \dots \cap r(T_k)$ for some trees T_1, T_2, \dots, T_k (Proposition 5).

Dekker [7] observed that if sets of rules are applied to a set of quartets and a contradiction results, then the set of quartets is inconsistent. This is also true for sets of rooted triples. We prove that, in the rooted triple case, if we apply all possible rules then the converse of Dekker's observation is also true (Proposition 9 (2) below).

Proposition 9 *1. If R is a closed set of rooted triples containing no triple with the leaves $\{a, b, c\}$ then $R \cup \{ab|c\}$, $R \cup \{ac|b\}$ and $R \cup \{bc|a\}$ are all consistent.*

2. If all possible rooted triple rules are applied to the consistent subsets of a set R of rooted triples then a contradiction (e.g. $ab|c$ AND $ac|b$) is derived if and only if the set R is inconsistent.

3. If R is a set of at least three rooted triples and every proper subset of R is consistent and closed, then R is consistent.

Proof

(1) Suppose that one of $R \cup \{ab|c\}$, $R \cup \{ac|b\}$ and $R \cup \{bc|a\}$ is inconsistent, say $R \cup \{ab|c\}$. By Lemma 2, there is $S : |S| \geq 3$ such that $[R, S]$ has exactly two components, with a and b in different components.

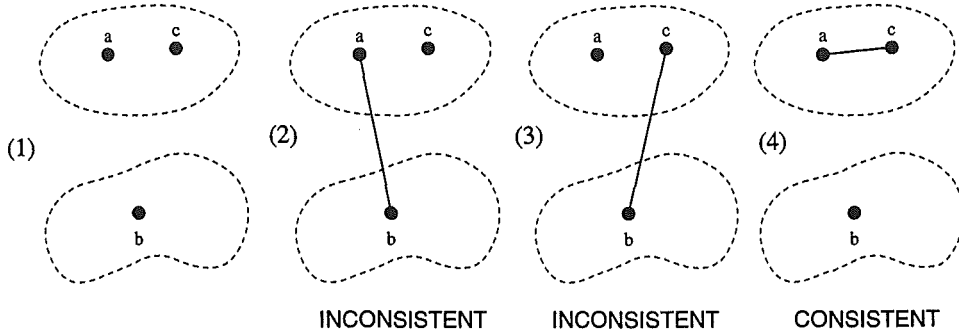


Figure 6 : The components of $[R, S]$. The dotted lines indicate the components of $[R, S]$ (1). If we add an edge between a and b , (2), or between c and b , (3), then we get a connected graph. Hence the only triple with these leaves that is consistent with R is $ac|b$ (4).

If c is in the same component as a then $[R \cup bc|a, S]$ is connected so $R \cup bc|a$ is inconsistent (Figure 6). Since $R \cup ab|c$ is also inconsistent we have, by elimination, that every tree compatible with R is also compatible with $ac|b$. That is, $ac|b \in \overline{R}$. Similarly, if c is in the same component as b then $bc|a \in \overline{R}$. In either case we obtain a contradiction.

(2) If R is consistent then \overline{R} is also consistent, so applying all possible rules to R will give a consistent set that contains no contradictions.

Conversely, suppose that R is inconsistent. Let R_c be a maximal consistent subset of R and choose $ab|c$ in $R \setminus R_c$. Then $\overline{R_c} \cup \{ab|c\}$ is inconsistent, so by (1) either $ac|b \in \overline{R_c}$ in which case $R_c \vdash ac|b$, or $bc|a \in \overline{R_c}$ and $R_c \vdash bc|a$. In both cases we derive a contradiction with $ab|c$.

(3) Let $ab|c \in R$. Then $R \setminus \{ab|c\}$ is consistent and closed. Every subset of R is consistent, so R contains at most one of $ab|c, ac|b, bc|a$. Hence $ac|b \notin R \setminus \{ab|c\}$ and $bc|a \notin R \setminus \{ab|c\}$ so $R \setminus \{ab|c\} \cup \{ab|c\}$ is consistent by part (1). \square

Note that (1) does not hold for the quartet case. For example, if

$$Q = \{12|36, 23|45, 14|56\}$$

then $Q \cup 13|24$ is inconsistent. We were unable to determine whether (2) was true for quartets.

Let R be a consistent set of m triples. The closure of R can be found in polynomial time. There are at most $3m$ different leaves in $\mathcal{L}(R)$. Consider each subset of $\mathcal{L}(R)$

with three leaves, say $\{a, b, c\}$. There are $O(m^3)$ such sets. Test the consistency of $R \cup \{ab|c\}$, of $R \cup \{ac|b\}$, and of $R \cup \{bc|a\}$ using the Algorithm ONETREE. If exactly one set is consistent then the corresponding triple is in \bar{R} , otherwise there is no triple in \bar{R} with leaves $\{a, b, c\}$. Checking each set of triples takes $O(m^2)$ time [2]. Hence the entire process takes $O(m^5)$ time. It is reasonable to expect that a far more efficient algorithm could be found — our aim here is simply to show that the problem can be solved in polynomial time.

We now characterize closed sets of rooted triples in terms of the graphical representation.

Proposition 10 *Let R be a consistent set of rooted triples. R is closed if and only if for each set S , $|S| \geq 3$ for which $[R, S]$ has exactly two components, these components are cliques and the label set of each edge contains every label in the other component.*

Proof

Suppose that R is closed. Let S be a subset of $\mathcal{L}(R)$ such that $[R, S]$ has two components. Choose any a and b in one component and any c in the other component. Both $[R \cup \{ac|b\}, S]$ and $[R \cup \{bc|a\}, S]$ are connected, so by Theorem 2, both $R \cup \{ac|b\}$ and $R \cup \{bc|a\}$ are inconsistent. By elimination $R \vdash ab|c$. Since R is closed, $ab|c \in R$ so there is an edge between a and b with c in its label set. The result follows.

Conversely suppose that R is not closed. There is $ab|c$ not contained in R , even though $R \vdash ab|c$. Now $R \cup ac|b$ must therefore be inconsistent, so by Lemma 2 there is $S \subseteq \mathcal{L}(R)$, $|S| \geq 3$, such that $[R, S]$ has two components, with a and b in one component and c in the other. But since $ab|c$ is not in R , the edge from a to b does not have c in its label set. \square

If a set R of rooted triples is consistent and closed, and every subset of R is also closed then we say that R is **fully closed**. A characterization of fully closed sets stems directly from the preceding proposition, as follows.

Proposition 11 *A consistent set R is fully closed if and only if for all $S \subseteq \mathcal{L}(R)$ with $|S| \geq 4$, the graph $[R, S]$ has at least three components.*

Proof

Suppose that for all $S \subseteq \mathcal{L}(R)$ with $|S| \geq 4$ the graph $[R, S]$ has at least three components. Let $R' \subseteq R$ and $S \subseteq \mathcal{L}(R)$ with $|S| \geq 3$. The graph $[R', S]$ is a subgraph of $[R, S]$. Now R is consistent so if $|S| = 3$ then either $[R', S]$ has no edges and hence 3 components, or $[R', S]$ has one edge, labelled by the vertex in the second component. By Proposition 10, R' is closed. We conclude that R is fully closed.

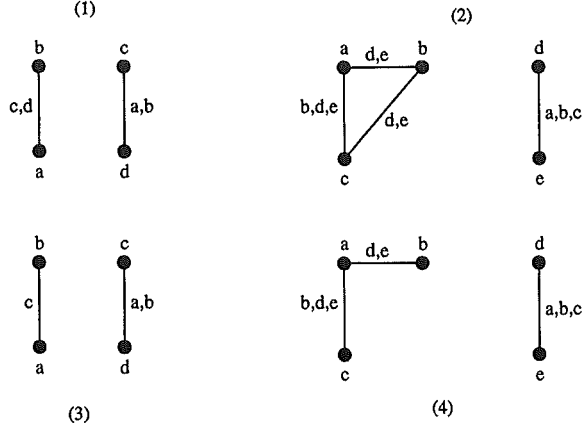


Figure 7 : If R is closed and $[R, S]$ has two components, then either both components have two vertices (1), or one component has three or more vertices (2). In (3) we have removed the triple $ab|d$ from R giving a set that is not closed. In (4) we have removed both $bc|d$ and $bc|e$, giving a subset of R that is not closed.

Conversely, let R be fully closed. Consider any subset $S \subseteq \mathcal{L}(R)$ with $|S| > 3$. R is consistent, so by Theorem 2, $[R, S]$ has at least two components. Suppose that $[R, S]$ has only two components. Either both components have exactly two vertices, or one component has at least three vertices. In the first case both components have exactly one edge and this edge is labelled by the two vertices in the other component (Figure 7 (1)). Removing one triple from R that has leaves in S will remove one of the labels from one of the edges, giving a set that is not closed by Proposition 10 (Figure 7 (3)). In the second case (Figure 7 (2)), removing an edge will still leave a graph with two components that corresponds to a subset of R that is not closed (Figure 7 (4)). In either case, R is not fully closed. \square

We apply these results to an example that we use in the next section.

Proposition 12 *Let $A = \{a_1, a_2, \dots, a_p\}$, $B = \{b_1, b_2, \dots, b_q\}$, and $C = \{c_1, c_2, \dots, c_r\}$, be disjoint sets of leaves. Let R be any set of rooted triples each of which are of the form $a_i a_j | b_k$ or $b_i b_j | c_k$ or $c_i c_j | a_k$, and which have the further property that for each $z \in A \cup B \cup C$, there is at most one triple in R of the form $xy|z$. Then R is consistent and fully closed.*

Proof

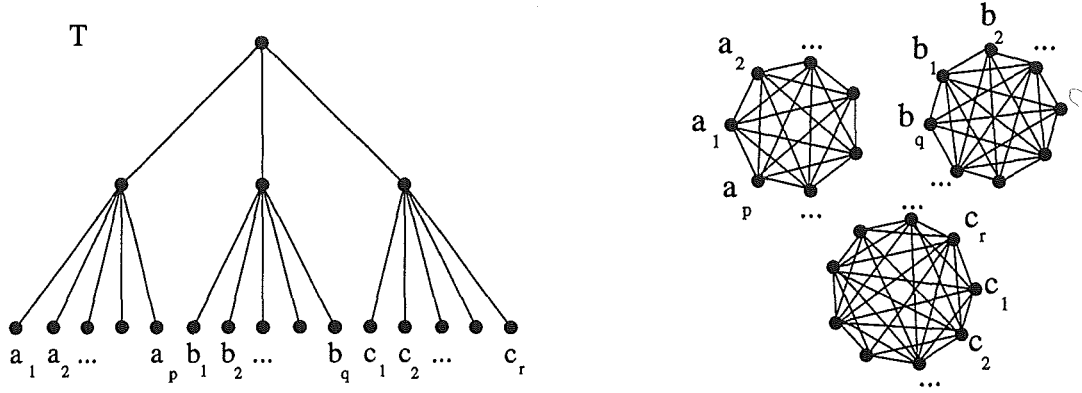


Figure 8 : The tree T on the left is compatible with the set R from Proposition 12, so that R is consistent. On the right is the graph $[r(T), \mathcal{L}(T)]$, consisting of three disjoint cliques on p , q and r vertices respectively. Note that $[R, S]$ is a subgraph of $[r(T), \mathcal{L}(T)]$ for any S (Proposition 8).

The tree T in Figure 8 is compatible with R , so R is consistent. We will use Proposition 11 to show that R is fully closed. Let S be a subset of $\mathcal{L}(R)$ with $|S| \geq 4$. Consider three cases.

Case 1: S contains at least one element from each of A , B and C .

Now $R \subseteq r(T)$ so $[R, S]$ is a subgraph of $[r(T), \mathcal{L}(T)]$ (Figure 8). The elements in A , B and C must be in different components of $[R, S]$. Therefore $[R, S]$ has at least three components.

Case 2 : S intersects exactly two of A, B and C .

By symmetry, we can assume without loss of generality that S is contained in $A \cup B$. The graph $[R, S]$ has at least two components because R is consistent (Theorem 2). Suppose that $[R, S]$ has only two components. As in Case 1, the elements in A and the elements in B are contained in different components of $[R, S]$. As $[R, S]$ has only two components, the vertices in $S \cap A$ are connected and the vertices of $S \cap B$ are connected.

If there is more than one vertex in $S \cap B$ then there is an edge in this component. However any such edge would be labelled by a vertex from C , giving a contradiction. On the other hand if there is only one vertex in the $S \cap B$ component, then there must be at least three vertices in the $S \cap A$ component, since $|S| \geq 4$. Hence there are at least two distinct edges in the $S \cap A$ component. We required R to have the property that for each z in $\mathcal{L}(R)$, there is at most one triple in R of the form $xy|z$. Each of these edges in $S \cap A$ are therefore labelled by a different element of $S \cap B$, a contradiction. We conclude that $[R, S]$ has at least three components.

Case 3: S is a subset of A , B or C .

Without loss of generality, assume that $S \subseteq A$. If $[R, S]$ has less than three components, then there must be an edge in $[R, S]$, simply because S has at least four elements. However all edges connecting vertices in A are labelled by vertices in B , so S must contain an element of B as well, a contradiction. In all three cases, R is fully closed by Proposition 11. \square

4 The existence of irreducible rules of arbitrarily high order

Dekker [7] describes a third order rule that cannot be derived through repeated application of second order rules. After studying rules with orders three, four and five he conjectures that for any n , there exist rules of order n that cannot be derived from rules involving fewer than n quartets. We prove this conjecture, first for rooted triples and then for quartets. Our strategy is to construct a set of n triples or quartets that is not closed even though every proper subset of it is closed.

We actually construct three sets of rooted triples.

$$\begin{aligned}
R_0 &:= \{a_1 a_2 | b_1, a_2 a_3 | b_2, \dots, a_m a_{m+1} | b_m, \\
&\quad b_1 b_2 | c_1, b_2 b_3 | c_2, \dots, b_{m-1} b_m | c_{m-1}, \\
&\quad c_1 c_2 | a_1, c_2 c_3 | a_2, \dots, c_m c_{m+1} | a_m, \\
&\quad a_{m+1} b_1 | c_{m+1}\}, & m \geq 1 \\
R_1 &:= \{a_1 a_2 | b_2, a_2 a_3 | b_3, \dots, a_m a_{m+1} | b_{m+1}, \\
&\quad b_1 b_2 | c_1, b_2 b_3 | c_2, \dots, b_m b_{m+1} | c_m, \\
&\quad c_1 c_2 | a_1, c_2 c_3 | a_2, \dots, c_m c_{m+1} | a_m, \\
&\quad a_{m+1} b_1 | c_{m+1}\}, & m \geq 1 \\
R_2 &:= \{a_1 a_2 | b_1, a_2 a_3 | b_2, \dots, a_m a_{m+1} | b_m, \\
&\quad b_1 b_2 | c_1, b_2 b_3 | c_2, \dots, b_{m-1} b_m | c_{m-1}, \\
&\quad c_1 c_2 | a_1, c_2 c_3 | a_2, \dots, c_{m-1} c_m | a_{m-1}, \\
&\quad a_{m+1} b_1 | c_m\}, & m \geq 2
\end{aligned}$$

Lemma 3 *For each $i = 0, 1, 2$, the set R_i is consistent. Furthermore, if S is a proper subset of $\mathcal{L}(R_i)$, $|S| \geq 4$, $i = 0, 1, 2$, then $[R_i, S]$ has at least three components.*

Proof

We prove the case of $i = 1$. The remaining cases are proved in a similar way. Let $R = R_1$. The tree in Figure 9 is compatible with R so R is consistent. Let $A = \{a_1, a_2, \dots, a_{m+1}\}$, $B = \{b_1, b_2, \dots, b_{m+1}\}$, $C = \{c_1, c_2, \dots, c_{m+1}\}$.

By Proposition 12, the set $R \setminus \{a_{m+1} b_1 | c_{m+1}\}$ is fully closed, so by Proposition 11, the graph $[R \setminus \{a_{m+1} b_1 | c_{m+1}\}, S]$ has at least three components. If one of a_{m+1}, b_1 or c_{m+1} is not in S then $[R, S]$ is the same graph as $[R \setminus \{a_{m+1} b_1 | c_{m+1}\}, S]$ and so also has three components.

Assume that $\{a_{m+1}, b_1, c_{m+1}\} \subset S$. Since $S \subseteq \mathcal{L}(R)$, the graph $[R, S]$ is a subgraph of $[R, \mathcal{L}(R)]$ (see Figure 9). The elements of $S \cap C$ and the elements of $S \cap (A \cup B)$ are in different components in $[R, \mathcal{L}(R)]$, so they are in different components of $[R, S]$. Because S contains elements of both $A \cup B$ and C , the graph $[R, S]$ has at least two components. Suppose that $[R, S]$ has only two components. Then all of the vertices in $S \cap (A \cup B)$ are connected, and all the vertices in $S \cap C$ are connected.

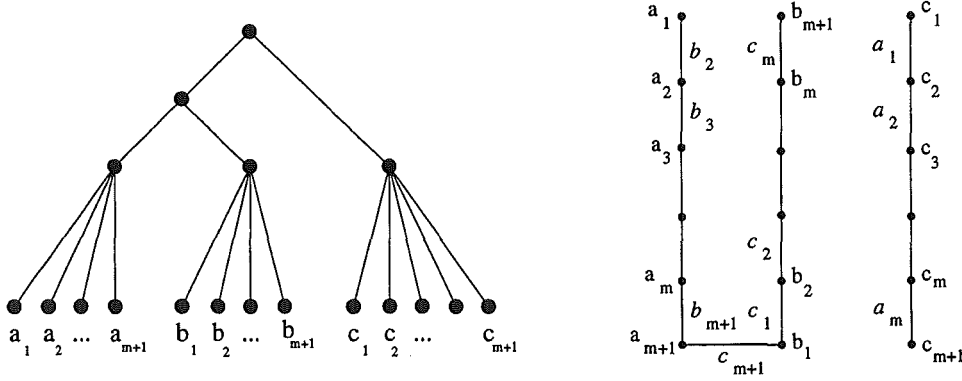


Figure 9 : The tree on the left is compatible with R , so that R is consistent. On the right is the graph $[R, \mathcal{L}(R)]$ and associated edge labelling.

There are at least four elements in S , so there is at least one additional element x in S other than a_{m+1}, b_1 and c_{m+1} . Consider the cases of $x \in A$, $x \in B$ and $x \in C$.

Case 1: $x \in A$

Let $x = a_i$. The vertices a_i and a_{m+1} are in the same component of $[R, S]$, so there is a path in $[R, S]$ going from a_i to a_{m+1} . Now the only path in $[R, \mathcal{L}(R)]$ (Figure 9) from a_i to a_{m+1} passes through a_i, a_{i+1}, \dots, a_m , and a_{m+1} . Since $[R, S]$ is a subgraph of $[R, \mathcal{L}(R)]$, the only possible path from a_i to a_{m+1} in $[R, S]$ passes through these same vertices. Therefore $a_i, a_{i+1}, \dots, a_m, a_{m+1}$ and the labels of the edges connecting them in $[R, S]$ are also in S . In particular the edge connecting a_m and a_{m+1} is in $[R, S]$, so $b_{m+1} \in S$.

But b_{m+1} is in the same component of $[R, S]$ as b_1 and a_i . Therefore there is a path in $[R, S]$ from b_{m+1} to b_1 . Referring to the graph $[R, \mathcal{L}(R)]$ we observe that the only path from b_{m+1} to b_1 in $[R, \mathcal{L}(R)]$, and hence in the subgraph $[R, S]$, passes through every vertex of B . Therefore all the vertices in B are also in S , as well as the labels of the edges connecting them in $[R, S]$. In particular the edge connecting b_1 and b_2 is in $[R, S]$, so $c_1 \in S$.

But c_1 is in the same component of $[R, S]$ as c_{m+1} . Therefore there is a path in $[R, S]$ from c_1 to c_{m+1} . All the vertices in C are also in S , as well as the labels of the edges

connecting them in $[R, S]$. In particular the edge connecting c_1 and c_2 is in $[R, S]$, so $a_1 \in S$. Therefore there is a path from a_1 to a_{m+1} in $[R, S]$ and all the vertices in A are also in S . We have now shown that $S = \mathcal{L}(R)$, giving a contradiction.

Case 2: $x \in C$

Let $x = c_i$. The vertices c_i and c_{m+1} are in the same component of $[R, S]$. Therefore there is a path in $[R, S]$ from c_i to c_{m+1} . This is only possible if c_i, c_{i+1}, \dots, c_m and the labels of the edges connecting them in $[R, S]$ are also in S . In particular the edge connecting c_m and c_{m+1} is in $[R, S]$, so $a_m \in S$. Hence a_{m+1} is not the only element of A in S . Referring to the first case we obtain a contradiction.

Case 3: $x \in B$

Let $x = b_i$. The vertices b_i and b_1 are in the same component of $[R, S]$. Therefore there is a path in $[R, S]$ from b_1 to b_i . This is only possible if $b_2, b_3, \dots, b_i - 1$ and the labels of the edges connecting them in $[R, S]$ are also in S . In particular the edge connecting b_1 and b_2 is in $[R, S]$, so $c_1 \in S$. Hence c_{m+1} is not the only element of C in S . Referring to the second case we obtain a contradiction. \square

We define the set $R(n)$ for $n > 3$, as follows

If $n \equiv 0 \pmod{3}$ then put $m = n/3$ and $R(n) = R_0$.

If $n \equiv 1 \pmod{3}$ then put $m = (n - 1)/3$ and $R(n) = R_1$.

If $n \equiv 2 \pmod{3}$ then put $m = (n + 1)/3$ and $R(n) = R_2$.

In all three cases, $R(n)$ has n triples.

Theorem 3 *Given any $n > 3$ there is a consistent set of n rooted triples that is not closed even though every proper subset is closed. Thus there is a rooted triple rule of order n that cannot be derived by repeated application of rules of order less than n .*

Proof

Put $R = R(n)$ and let R' be any proper subset of R . We use Proposition 11 to prove that R' is fully closed. Let $S \subseteq \mathcal{L}(R') : |S| \geq 4$. Note that $\mathcal{L}(R') \subseteq \mathcal{L}(R)$.

Suppose that $\mathcal{L}(R') \neq \mathcal{L}(R)$. By Lemma 3 the graph $[R, S]$ has at least three components. Now $[R', S]$ is a subgraph of $[R, S]$ with the same vertices, so it must also have at least three components.

In a similar way, if $\mathcal{L}(R') = \mathcal{L}(R)$ and $S \neq \mathcal{L}(R')$ then $[R', S]$ has at least three components.

Finally, if $\mathcal{L}(R') = \mathcal{L}(R)$ and $S = \mathcal{L}(R')$ then $[R', S]$ is a subgraph of $[R, \mathcal{L}(R)]$ with the same vertices. Examining the diagram of $[R, \mathcal{L}(R)]$ in Figure 9 reveals that any such subgraph has at least three components.

By Proposition 11, R' is fully closed. Now R is consistent by Lemma 3. We show that R is *not* closed. The graphs $[R \cup \{a_1c_1|b_1\}, \mathcal{L}(R)]$ and $[R \cup \{b_1c_1|a_1\}, \mathcal{L}(R)]$

are both connected so by Theorem 2 the sets $R \cup \{a_1c_1|b_1\}$ and $R \cup \{b_1c_1|a_1\}$ are inconsistent. Hence $R \vdash a_1b_1|c_1$, even though $a_1b_1|c_1 \notin R$.

It follows that $R \vdash a_1b_1|c_1$ is a rule that cannot be reduced to repeated application of rules to subsets of R .

Theorem 3 can be reformulated in terms of quartets, thereby proving Dekker's original conjecture [7]. First we introduce an important link between sets of rooted triples and sets of quartets.

Suppose we have a rooted tree T . Let T_ρ denote the unrooted tree obtained by adding a new leaf adjacent to the root. This procedure gives a bijection between rooted trees and unrooted trees with a leaf ρ . Using the same principle we can convert a rooted triple $ab|c$ into a quartet $ab|c\rho$. In fact a rooted tree T is compatible with a rooted triple $ab|c$ if and only if T_ρ is compatible with $ab|c\rho$. This correspondence has a number of useful properties.

Proposition 13 *Let R be a set of rooted triples. Let Q be the associated set of quartets:*

$$Q := \{xy|z\rho : xy|z \in R\}$$

then

1. $\langle Q \rangle = \{T_\rho : T \in \langle R \rangle\}$
2. Q is consistent if and only if R is consistent.
3. If Q is closed then R is closed.
4. If $Q \vdash ab|c\rho$ then $R \vdash ab|c$

Proof

For (1) and (2) we observe that a tree T is compatible with R if and only if T_ρ is compatible with Q .

(3) If Q is closed then, by Proposition 4, there are binary trees $T_\rho^1, \dots, T_\rho^k$ such that

$$Q = q(T_\rho^1) \cap \dots \cap q(T_\rho^k).$$

Now, for any rooted tree T , $r(T) = \{ab|c : ab|c\rho \in q(T_\rho)\}$. Because all the quartets in $q(T_\rho^1) \cap \dots \cap q(T_\rho^k)$ share leaf ρ it follows that $Q = \{ab|c\rho : ab|c \in r(T^1) \cap \dots \cap r(T^k)\}$. Hence

$$R = r(T^1) \cap \dots \cap r(T^k)$$

and so R is closed by Proposition 4.

(4) If $Q \vdash ab|c\rho$ then $ab|c\rho \in q(T_\rho)$ for all $T_\rho \in \langle Q \rangle$. Hence by (1), $ab|c \in r(T)$,

$\forall T \in \langle R \rangle$, and so $R \vdash ab|c$.

Thus, if a set of quartets all share one leaf, one can convert the set into a corresponding set of rooted triples and determine in polynomial time, whether or not the quartets are consistent.

To extend Theorem 3 to quartets, we take the set of rooted triples R_i used to prove the rooted triple case, and convert it into a set of quartets Q_i , as described in Proposition 13. Unfortunately the unrooted analogue of Theorem 3 does not follow directly because the converse of Proposition 13 (3) is **not** true. For example, $\{ab|c, ab|d\}$ is a fully closed set of rooted triples, but $\{ab|c\rho, ab|d\rho\}$ is not a closed set of quartets.

Instead we use a further property of rooted triples.

Lemma 4 *Let R be any consistent set of rooted triples and suppose that $R \cup \{ab|c\}$ and $R \cup \{ab|d\}$ are both consistent. Then $R \cup \{ab|c, ab|d\}$ is also consistent.*

Proof

Let S be any subset of $\mathcal{L}(R) \cup \{a, b, c, d\}$. The sets $R \cup \{ab|c\}$ and $R \cup \{ab|d\}$ are both consistent, so by Theorem 2 the graph $[R \cup \{ab|c\}, S]$ and the graph $[R \cup \{ab|d\}, S]$ are both disconnected.

If $a \notin S$ or $b \notin S$ then the graph $[R \cup \{ab|c, ab|d\}, S]$ is the same as the graph $[R, S]$, so is disconnected (Theorem 2).

If $c \notin S$ then the graph $[R \cup \{ab|c, ab|d\}, S]$ is the same as the graph $[R \cup \{ab|d\}, S]$, so is disconnected. By symmetry, if $d \notin S$ then $[R \cup \{ab|c, ab|d\}, S]$ is disconnected.

Finally, if a, b, c and d are all in S , then the graph $[R \cup \{ab|c, ab|d\}, S]$ is the same as the graph $[R \cup \{ab|c\}, S]$ with an extra label d on the edge (a, b) , so the graph is still disconnected. In any of these five cases, the graph $[R \cup \{ab|c, ab|d\}, S]$ is disconnected. Hence $R \cup \{ab|c, ab|d\}$ is consistent, by Theorem 2. \square

Theorem 4 *Given any n there is a consistent set of n quartets that is not closed even though every proper subset is closed. Thus there is a quartet rule of order n that cannot be derived by repeated application of rules of order less than n .*

Proof

When $n = 1, 2$ the proof is trivial. If $n = 3$ then a suitable example is $\{ab|cd, ab|ef, ce|df\}$ [7].

When $n > 3$, let $R = R(n)$, where $R(n)$ is defined just before Theorem 3, and construct the set of quartets

$$Q := \{xy|z\rho : xy|z \in R\}.$$

For example, when $n \equiv 1 \pmod{3}$:

$$\begin{aligned} Q := & \{a_1a_2|b_2\rho, a_2a_3|b_3\rho, \dots, a_ma_{m+1}|b_{m+1}\rho, \\ & b_1b_2|c_1\rho, b_2b_3|c_2\rho, \dots, b_mb_{m+1}|c_m\rho, \\ & c_1c_2|a_1\rho, c_2c_3|a_2\rho, \dots, c_mc_{m+1}|a_m\rho, \\ & a_{m+1}b_1|c_{m+1}\rho\} \end{aligned}$$

By Proposition 13, Q is closed. We claim that every proper subset of Q is closed.

Let Q' be a proper subset of Q and let R' be the corresponding subset of R . Consider any four leaves a, b, c and d in $\mathcal{L}(R')$. No two triples in R have more than one leaf in common so there is at most one triple in R , and therefore in R' , with all its leaves in $\{a, b, c, d\}$. If there is such a triple in R' we assume, without loss of generality, that this is the triple $ab|c$. Hence there are no triples in R with leaves $\{a, c, d\}$, $\{b, c, d\}$ or $\{a, b, d\}$. Of course this also applies if there is no triple in R' with leaves in $\{a, b, c, d\}$. By Proposition 9 (1) we have

$$(i) \ R' \cup \{cd|a\}, R' \cup \{cd|b\} \text{ are both consistent}$$

and

$$(ii) \ R' \cup \{ad|c\}, R' \cup \{ad|b\} \text{ are both consistent}$$

Applying Lemma 4 to (i), the set $R' \cup \{cd|a, cd|b\}$ is consistent, so by Proposition 13 (2), the set $Q' \cup \{cd|a\rho, cd|b\rho\}$ is consistent. But $\{cd|a\rho, cd|b\rho\} \vdash cd|ab$, so $Q' \cup \{cd|ab\}$ is consistent.

By Lemma 4 and (ii), the set $R' \cup \{ad|c, ad|b\}$ is consistent, so by Proposition 13 the set $Q' \cup \{ad|c\rho, ad|b\rho\}$ is consistent. But $\{ad|c\rho, ad|b\rho\} \vdash ad|bc$, so $Q' \cup \{ad|cb\}$ is consistent. Hence there is no quartet with leaves $\{a, b, c, d\}$ in the closure of Q' .

Thus, to prove that Q' is closed we only need to show now that there are no quartets of the form $ab|c\rho$ in the closure of Q' that are not already contained in Q' . If $Q' \vdash ab|c\rho$ then by Proposition 13 (4), $R' \vdash ab|c$. As R' is closed, this implies that $ab|c \in R'$, and so $ab|c\rho \in Q'$.

Hence Q' is closed. We show that Q itself is not closed. Recall from the proof of Theorem 3 that $R \vdash a_1b_1|c_1$. By Proposition 13 (4), $Q \vdash a_1b_1|c_1\rho$. It follows that $Q \vdash a_1b_1|c_1\rho$ is a rule of order n that cannot be derived through repeated application of rules with order less than n .

Remark

An earlier attempt at proving Theorem 3 led to a related result, that for any $k \geq 1$ there exists a set of rooted triples that is inconsistent even though every subset of

size at most k is consistent and closed. Of course, by Proposition 9 (3), if *every* proper subset of a set of rooted triples is consistent and closed then the entire set is consistent, so we cannot expect a full analogue of Theorem 3 to apply here.

Let $m = 3k + 1$ and put

$$\begin{aligned} R_3 := \{ & a_1 a_2 | b_1, a_2 a_3 | b_2, \dots, a_{m-1} a_m | b_{m-1}, \\ & b_1 b_2 | c_1, b_2 b_3 | c_2, \dots, b_{m-1} b_m | c_{m-1}, \\ & c_1 c_2 | a_1, c_2 c_3 | a_2, \dots, c_{m-1} c_m | a_{m-1}, \\ & a_m b_1 | b_m, b_m c_1 | c_m \} \end{aligned}$$

The structure of R_3 is revealed by the associated graph $[R_3, \mathcal{L}(R_3)]$, represented in Figure 10.

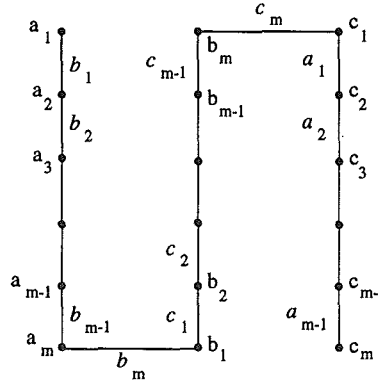


Figure 10 : The graph $[R_3, \mathcal{L}(R_3)]$. The edges are labelled in italics. The graph is connected, so R_3 is inconsistent by Theorem 2.

Using arguments similar to the proofs of Lemma 3 and Theorem 3, it can be shown that every subset R' of R_3 with $|\mathcal{L}(R')| < m$ is both consistent and closed. Hence every subset of R_3 with k or fewer triples is also consistent and closed, and yet the set R_3 is inconsistent, by Theorem 2 (and Figure 10). It follows that the set of rooted triple rules of order k or less is insufficient to determine not only the closure of a set (Theorem 3), but also the consistency of a set. This proves another conjecture of [7].

References

- [1] E.N. ADAMS, N-trees as nestings: Complexity, similarity and consensus, *J. Classification* **3** (1986), 299-317.
- [2] A.V. AHO, Y. SAGIV, T.G. SZYMANSKI AND J.D. ULLMAN, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, *SIAM J. Comput.* **10** No. 3 (1981), 405-421.
- [3] A. AMIR AND D. KESELMAN, Maximum agreement subtree in a set of evolutionary trees, *Proc. 35th IEEE FOCS, Santa Fe* (1994).
- [4] H.-J. BANDELT AND A.W.M. DRESS, Reconstructing the shape of a tree from observed dissimilarity data, *Advances in Applied Mathematics* **7** (1986), 309-343.
- [5] R.L. CANN, M. STONEKING AND A.C. WILSON, Mitochondrial DNA and human evolution, *Nature* **325** (1987), 31-36.
- [6] M. CONSTANTINESCU AND D. SANKOFF, An efficient algorithm for supertrees, *J. Classification* (1994) (in press).
- [7] M.C.H. DEKKER, "Reconstruction methods for derivation trees", Masters Thesis, University of Amsterdam, 1986.
- [8] A.W.M. DRESS AND M.A. STEEL, Convex tree realizations of partitions, *Appl. Math. Lett.* **5**, No. 3 (1992), 3-6.
- [9] N. ELDREDGE AND J. CRACRAFT. "Phylogenetic Patterns and the Evolutionary Process," Columbia University Press, New York, 1980.
- [10] S.K. KANNAN AND T.J. WARNOW, Inferring Evolutionary History from DNA Sequences, *SIAM J. Comput.* **23**, No. 4 (1994), 713-737.
- [11] DAVID R. MADDISON, African Origin of Human Mitochondrial DNA Reexamined, *Syst. Zool.* **40**, No. 3 (1991), 355-393.
- [12] M.P. NG AND N.C. WORMALD, Reconstruction of rooted trees from subtrees, *Discr. Appl. Math.* (1994) (in press).
- [13] M.A. STEEL, "Distributions on bicoloured evolutionary trees", Ph.D. thesis, Massey University, Palmerston North, New Zealand, 1989.
- [14] M.A. STEEL, The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classification* **9** (1992), 91-116.

- [15] M.A. STEEL, Decompositions of leaf-coloured binary trees, *Adv. Appl. Math.* **14** (1993), 1-24
- [16] W. VACH, Preserving Consensus Hierarchies, *J. Classification* **11** (1994), 59-77.
- [17] T.J. WARNOW, "Combinatorial algorithms for constructing phylogenetic trees," Ph.D. thesis, University of California, Berkeley, CA, USA, 1991.
- [18] T.J. WARNOW, Tree compatibility and inferring evolutionary history, *Journal of Algorithms* **16** (1994), 388-407.
- [19] D.J.A. WELSH, "Matroid Theory", Academic Press Inc. (London) Ltd., London, 1976.
- [20] MARK WILKINSON, Common cladistic information and its consensus representation: reduced Adams and reduced cladistic consensus trees and profiles, *Syst. Biol.* **43**, No. 3 (1994), 343-368.